RESEARCH ARTICLE OPEN ACCESS

Importance of Agile Methodology in Software Development

Harihar Sahoo*, Sabyasachi Pattnaik**
*(Department of I&CT, Fakir Mohan University, Balasore
Email: harihar2008@gmail.com)
** (Department of I&CT, Fakir Mohan University, Balasore
Email: spattnaik40@yahoo.co.in)

----------**********----------

Abstract:

The agile software development methodology has recently become one of the most commonly used software development techniques. Rather than the long drawn out release cycles in the previously popular waterfall methodology, the agile technique suggests regular short sprint release cycles. This allows the customers and stakeholders to have more involvement within the software development process. This helps promote a higher quality final product because it combats the difficult task of a customer fully understanding and identifying all requirements in the software project planning phase. Software testing is the most significant practice to verify the quality of a product. Software testing in Agile development is very complicated and controversial issue in literature and industry. Different people have different views about software testing in Agile methods, because most of Agile methods do not focus much on software testing activities. Agile strongly focus on the close customer collaboration, short iterations and frequent deliveries. But when it comes to software testing, then it is challenging, as Agile do not include many destructive testing practices, which are normally required for a quality product. This present work also identifies the practices of Agile development in industry and the critical issues in industry while practicing Agile development. The issues of automated and manual testing, good practices in automation, and how to manage independent testing teams in Agile development are also high lightened. In this paper, we also highlight every aspect of software testing process in Agile development according to the literature reviews and an industrial survey.

Keywords — **Software testing, Agile methodology, Quality assurance.**

----------**********----------

I. INTRODUCTION

Software Testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system. Software development is really a complex task. There are so many extensive methodologies that have been developed to give a step by step guidance to the organizations, while developing a system. Waterfall approach is a conventional way of developing software. It is a traditional way of developing software. In this approach some predefined steps are being followed like; requirements, design, coding, testing, and maintenance. The requirements for the system are fully defined at the start of the process, then a design is fully created, then coding and testing of the system occurs. Each of these activities leads to the next. This kind of heavyweight methodology contains a huge documentation for every step. The project team is expected to follow the structured plan, requirements are gathered at the start and then work separately from the customer, and finally deliver the complete piece of software that meets the initial expectations by keeping in mind the cost and the total time frame. Figure 1 shows the

waterfall process for software development. It shows the flow of different phases during development. At first the requirements for the system are gathered and they develop a requirements specification document.

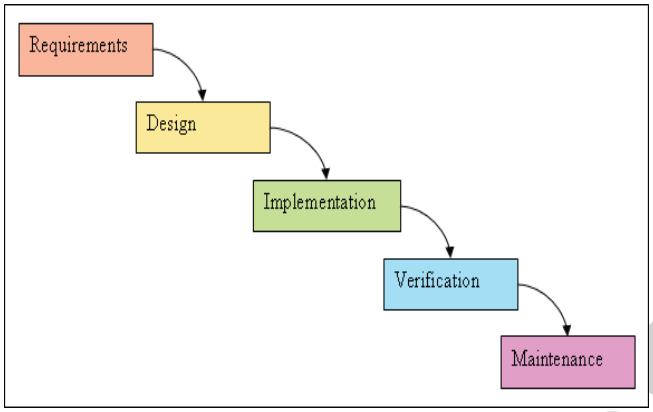

Fig. 1 Waterfall process for software development

The iterative approach has become vastly effective in helping software developers improve their skills in estimating schedule for remaining tasks. Schedule estimation is one of the most difficult responsibilities for developers because software issues are common and are unpredictable by nature. By breaking the large requirements down into more manageable sub requirements, the agile process naturally promotes better estimation.

In today's software development world, it is becoming more important than ever to keep the software stakeholders as involved as possible. With any software project there can be countless stakeholders who all can affect or can be affected by the outcome of the software. It is important for the software development team to identify the important stakeholders and find ways to connect with them individually to help promote stakeholder interest and involvement in the product. As agile promotes constant short release cycles, the stakeholders can see continual progress and make suggestions and develop new improved ideas for the software.

II. AGILE SOFTWARE DEVELOPMENT

In response to these conventional and rigid methodologies, iterative development processes were born which are called as Agile. Agile allows change in the requirements throughout the development cycle and stresses on the close coordination between developer and customer. The central idea is the close involvement and a frequent communication between the development team and stakeholders and delivery of functionality on a regular base. Agile is a flexible approach to development. According to [1], Agile gives preferences to the individuals and the interactions among them over processes and tools. Agile methodologies are formed on a concept that the individuals working in the organization are the most important part of the project. There should be proper communication between the team members. Because, if the communication among the team members will be regular then they will be able to overcome some of the important problems and there will be more chances for individuals to learn from the experiences of their senior members. In Agile, customer actively participates in the whole development process and guides the team about the system's requirements. Agile methodologies prefer a quick response to change over following some predefined plan.

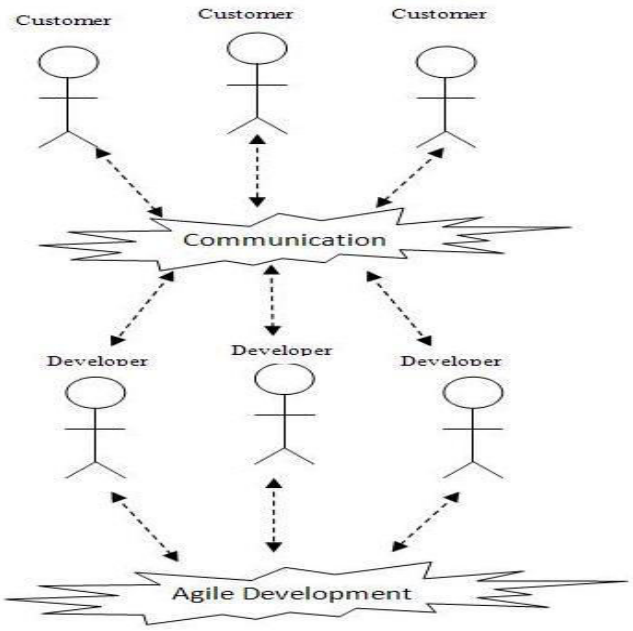

Fig. 2 The Agile working

Figure 2 shows the working of Agile process. In Agile the developers work very closely to the customers and the most important part of the

process is customer. Developers interact directly with customers to get feedback and to deliver a working product. Developer discusses each and everything with customer and prioritizes his/her work. The main focus of Agile is the satisfaction of the customer through a quick and continuous delivery of useful and working pieces of software".

A. *Principles of Agile*

Agile practitioners keep in mind different ways or basic set of rules during any of Agile methodology. According to [15], following are the principles of Agile development:

1. Satisfy the customer through the early and quick delivery.
2. Welcome change in requirements even in the late in the project.
3. Keep delivery cycle short (weeks).
4. Business and development people should work together.
5. Build project around some motivated people.
6. Place emphasis on face to face communication.
7. Working software is primary measure of progress.
8. Promote substantial development pace.
9. Continuous attention to the good design and technical excellence.
10. Simplicity is essential.
11. Best result comes from self organizing team.
12. Teams discuss regularly that where and how to improve.

There are various Agile methods like; Scrum, Crystal Clear, Extreme Programming, Adaptive Software Development, Feature Driven Development and DSDM (Dynamic Systems Development Method). The most common of them are XP and Scrum. XP or Extreme Programming has four key values including simplicity, communication, courage and feedback. It has small releases pair programming and delivering of business values. The Dynamic Systems Development Method also calls for the involvement of the customer, scheduled delivery of project chunks and iterative development cycles. crum begins with the making of a product backlog that is a list of customer requirements. Then they prioritize each element of the backlog to know that which item should be given the higher attention and work on the highest first. Here, each iteration is a *Sprint* that consists of a period of one month's duration. There is proper planning at the start of each sprint and every day starts with a 15 minutes meeting, where they discuss individuals work and to discuss their daily task. The frequent communication helps the development process easily adopt the changes in priorities and content. Team presents the functionality to the product owner for review at the end of each sprint.

[2] Says that Agile methods such as Extreme Programming, Crystal, and Scrum etc. have got a great attention recently. Agile methods focus on early and continuous delivery of software always welcome the changing requirements and give value to early feedback from customers. According to [3], Agile methodologies like SCRUM, XP are most suitable for companies with small or medium sizes because it may be hard for the management to handle larger teams with bigger numbers of individuals. Agile can be useful for the companies to achieve different factors like: Communication, estimation skills, iteration planning and responsibility [4].

B. *Automated Testing*

Now a day most people prefer to have automated testing but still there is need of manual testing to get rid of more bugs. It gives the ability to main stream scenario and run automation against code that frequently changes to catch regressions in a timely manner. The cost of automation is more especially when writing the tests or configuring the automate framework. The visual reference cannot be automated e.g. if the font color or size can't be defined via code then it is manual test. If a test case runs only twice then it should be manual to save the cost.

III. METHODOLOGY

Different methodologies of Agile development are as follows:

- Scrum
- Extreme programming (Xp)
- Feature Driven Development (FDD)
- Crystal Clear Methodology (CC)
- Dynamic Systems Development Method (DSDM)

- Adaptive Software Development

But first four methods are more popular and are being used in industry now days. Therefore we will discuss the practices of Scrum, Xp, FDD and CC in this paper. Most of the companies are working with Scrum methodology of Agile development in combination with Xp, so we felt that it will be not feasible for us to explain all of the practices of all methodologies.

SCRUM:

The SCRUM term is derived from the strategy in the game of rugby. In the game it means, getting an out of play ball back into the game with a team work. The Scrum approach has been to develop to manage the process for system development. This approach does not concentrate on some specific software development techniques for implementation phase. It basically guides the management that how team members can function in orders to achieve the system flexibility in an environment where the requirements are constantly changing. The basic idea of Scrum is that; there are several technical and environmental variables that are likely to change during the process. These variables include requirements, resources, time and technology.

EXTREME PROGRAMMING:

Extreme programming (Xp) has been introduced as a result to the long development cycles of tradition and conventional development models. It is one of the known methodologies of Agile development. It is a collection of different practices, rules and ides that are inherited from some previous methodologies. The main characteristics of Xp are short iterations with rapid feedback and small releases, the regular participation of customer, continuous testing and integration activities, collective ownership of the code, not a huge amount of documentation and pair programming. In Xp the physical environment is really important, and Xp is more effective in the companies with smaller and medium sizes.

FEATURE DRIVEN DEVELOPMENT:

Feature-Driven Development (FDD) is a client-centric, architecture-centric, and pragmatic software process. The term "client" in FDD is used to represent what Agile Modeling (AM) refers to as project stakeholders or Xtreme Programming (Xp) calls customers. FDD was first introduced to the world in 1999 via the book Java Modeling in Color with UML. Feature driven development is essentially a software management method instead a software engineering life cycle development. FDD is less Agile because it uses many established software development methods. FDD involves planning and up-front modeling and design.

CRYSTAL CLEAR METHODOLOGY:

Crystal Clear is the smallest of a series of methodologies for software development. An Agile methodology which is used with small teams i.e. 2-8 persons. According to Alistair *Cockburn*, it attempts to describe a full methodology of the lightest, most habitable kind that will still produce good results. It is prioritized for *project safety* (delivering a system in adequate time / budget per the sponsor's priorities) and for that to be effective.

IV. CHALLENGES IN SOFTWARE TESTING PROCESS IN AGILE

Agile manifesto is the set of rules or principles for Agile software development. These principles consist of the ideas that are basic guidelines and are common for all Agile development methods. If we take a look at the software testing process in Agile methods, we see that it's really different from the testing process in the conventional way of development; and from the traditional point of view the basic rules in Agile manifesto has some bigger challenges. First of all, the ultimate priority of Agile development is to deliver a working piece of software to customers early and continuously with a rapid releasing cycle. For testing process it is a challenge because, if the release cycles are rapid then it will put fixed deadlines for testing activities and that does not allow maximizing the testing time if more defects are found than estimated [5]. Secondly, Agile demands that changing requirements should be welcomed even in later stages of the development. Testing activities are traditionally been based on specifications that are completed in a phase of development and then they

can be used as a basis for test design and other quality assurance activities. And if they will change the requirements and which will eventually change these documents then it will challenge the traditional way of doing testing. The face to face conversation of developer and the customer also creates some challenges for testing activities [5].

V. CONCLUSIONS

The agile software development methodology is being widely accepted within the software development community. Agile attempts to simplify the software planning and estimation process by decomposing large requirements into small individual tasks. Analyzing small tasks allow the software development team to more accurately predict the level of effort required in order to implement the change. This allows the project manager to accurately depict the percentage complete of the software which allows them to continually track overall project progress against the originally planned progress. The agile process also is designed to help train developer in their schedule estimating skills throughout the lifecycle.

REFERENCES

[1] H. Jim, Agile Project Management: Creating Innovative Products, Addison- Wesley Professional, 2004.

[2] W. F. Tichy, Agile Development: Evaluation and Experience, In *Proceedings of the 26th international Conference on Software Engineering,* IEEE Computer Society, 2004.

[3] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, Agile Software Development Methods, Review and Analysis, VTT Publications, 2002.

[4] J. Highsmith, Agile Project Management: Creating Innovative Products, Addison Wesley Longman Publishing Co., 2004.

[5] I. Juha, R. Kristian, and L. Casper, Towards Understanding Quality Assurance in Agile Software Development, International Conference on Agility Management 2005.

[6] M. Pyhajarvi, K. Rautiainen, Integrating Testing and Implementation into Development,Engineering Management Journal, 2004,vol.16.

[7] M. Puleio, How Not to Do Agile Testing, In Proceedings of the Conference on AGILE,IEEE Computer Society, 2006.

[8] Serena, An introduction to Agile software development, Serena Software Inc., 2007.

[9] Conchango, The Agile revolution: Reduce project failure rates through adoption of Agile, URL: www.conchango.com,2008.

[10] S. R. Palmer and M . Felsing, A Practical Guide to Feature-Driven Development. Pearson Education, 2001.

[11] B. Fitzgerald, K. J. Stol, Continuous software engineering: a roadmap and agenda, J Syst Softw.,2015.

[12] K. Conboy, Agility from first principles: reconstructing the concept of agility in information systems development, Inf Syst Res, 2009.

[13] J. Bosch, the future of software engineering, IEEE Softw., 2016.

[14] P. Rodríguez, Continuous deployment of software intensive products and services: a systematic mapping study, J. Syst. Softw. 2016.

[15] P. Abrahamsson, Agile software development: Introduction, currant status and future, VTT Publications, 2005.